

Instruktion för instuderingsmaterial

Fakta

Kurs: Systemutvecklare C/C++ Extended 2024
Klass: SUVx24
Teknikområde: Programspråket C, Iteration

Learning Target

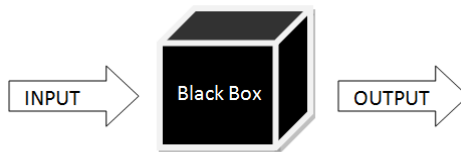
Programmering i C och C++

Hur man kan använda funktioner (**functions**) på olika sätt i C.

Introduktion

Att skriva program som en lösning på ett problem, leder ofta till många rader kod. Ett sätt att korta ner mängden kod i ett avsnitt, är att bryta ut den och lägga blocket i en separat del. Detta kan lösas med hjälp av funktioner i C.

En annan vanlig användning av funktioner är att se den som en svart låda. Ge lådan ett stimuli i form av indata (input), låt lådan (funktionen) bearbeta indata, lådan ger sedan feedback genom att returnera utdata (output) från lådan. Detta mönster används flitigt i programspråk.



Om vi använder en funktion som inte returnerar något tillbaka, kan vi i stället kalla det för en procedur. Den bara utför något, men ska inte lämna tillbaka något som resultat av bearbetningen. Dock kan proceduren mycket väl utföra visuell feedback som output. Detta sätt att använda funktion gör det möjligt att partitionera koden i lämpliga delar.

Material

Här nedan hittar du ditt självstudiematerial. Du ska studera materialet i den ordning det listas för att du på bästa sätt ska tillgodogöra dig materialet.

Instruktion för instuderingsmaterial

Learning Target

Hur man använder funktioner (**functions**) på olika sätt i C.

Hur

Läs nedanstående textavsnitt och de specificerade underdelarna. Studera kodexemplen för att förstå principerna.

Textavsnitt från [w3schools](#)

Funktioner har några mönster för deklaration som vi måste följa.

C Functions

- Läs avsnitten *Functions*, *Function Parameters*, *Scope*, *Function Declaration*, *Math Functions*.
- Vad används **functions** till?
- Vilka fördefinierade funktioner har du redan använt?
 - Var finns de fördefinierade funktionerna?
- Hur ser en funktionsdeklaration ut?
- Vad innebär ett anrop till en funktion?
- Vad är det för skillnad på parameter och argument?
- Hur ser en parameterlista ut?
- Vad innebär begreppet **scope**?
 - Studera betydelsen av räckvidd för en variabel.
 - Vilka scope finns det?
 - Hur kan samma variabelnamn deklarerats flera gånger i samma programfil?
- Vad innebär en funktions deklaration respektive definition?
 - Varför vill vi dela på dem?
 - Hur ser 'best practice' ut för detta?
- Studera var du kan hitta information om **C Standard Library** `<math.h>`

Instruktion för instuderingsmaterial

Textavsnitt från [geeksforgeeks.org](https://www.geeksforgeeks.org)

GeeksForGeeks förklarar functions med hjälp av bilder också.

C Functions

Förstå deklaration (**declaration**), definition och anrop (**calls**) av en funktion (**function**).

- På vilka fyra sätt kan man använda returtyp och parametrar (arguments)?
- Förstå skillnaden mellan fördefinierad funktion (Library Function) och användardefinierad (User Defined) funktion.
- Vilka sätt kan du skicka med argument till en funktion?
- Vad finns det för fördelar och nackdelar med funktioner?

Textavsnitt från [programiz.com](https://www.programiz.com)

C Functions

Ytterligare länkar med mer material kring funktioner.

- Läs avsnitten
 - *C Functions*
 - *C User-defined functions*
 - *Type of Used-defined Functions in C Programming*
- Vad menas med function prototype?
- Vad gör `return` sist i ett statement?
- Hur returneras värden **från** en funktion?
- Studera vilka typer av användardefinierad funktionsdeklaration.
 - Identifiera mönstren som kan användas och i vilka situationer du behöver dem.
- Titta gärna på videon längst ned på sidan.